

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

Bakalářská práce

2015

Tomáš Díčík

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Zadání bakalářské práce

Student:

Tomáš Díčík

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: VÍTKOVICE IT SOLUTION a.s.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. RNDr. Petr Šaloun, Ph.D.**

Konzultant bakalářské práce: Mgr. David Krčmarský

Datum zadání: 01.09.2014

Datum odevzdání: 07.05.2015



doc. Dr. Ing. Eduard Sojka
vedoucí katedry

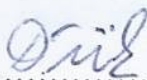


prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne **17.4.2015**


.....

Tomáš Díčík

Abstrakt

Práce popisuje mé zkušenosti a vědomosti získané na studentské stáži ve firmě Vítkovice IT Solutions a.s, kde jsem působil jako C# vývojář.

Jako vývojové prostředí bylo firmou používáno Microsoft Visual Studio, které firma opatřila každému vývojáři firmy. Později v práci je ukázka a detailnější popis práce v tomto vývojovém prostředí na firemním projektu pro záchrannou službu.

Dále je v práci popsána softwarová a hardwarová struktura informačního systému pro výše zmíněnou záchrannou službu. Pochopení struktury je důležité jak pro mě, vývojáře, podílející se na vývoji projektu, tak i pro čtenáře pročítající si práci.

V poslední řadě je popsána struktura a funkce komponenty informačního systému, kterou jsem měl na starosti. Komponenta slouží pro případný update určitých částí softwarové struktury informačního systému.

Klíčová slova

C# skript, T-SQL skript, update, komponenta, Microsoft Visual Studio, TFS - team foundation server, struktura, klient, server, design, windows služba

Abstract

This work describes my experience and knowledge gained on a student internship in a company Vitkovice IT Solutions Inc., where I worked as a C # developer.

As a development environment was used Microsoft Visual Studio, which the company sought for every developer in company. Later work is an example of a more detailed description of work in this development environment on company project for emergency services.

The work also describes the software and hardware structure of the information system for the aforementioned emergency services. Understanding this structure is important for me, developer, who is involved in the development of this project, as well as for the reader, who reads this work.

Finally my work describes the structure and function of component of the information system, which I was in charge. Component serves for possible update for individual parts of information system.

Keywords

C# script, T-SQL script, update, component, Microsoft Visual Studio, TFS - team foundation server, structure, client, server, design, windows service

Obsah

| | |
|--|----|
| 1. Úvod | 8 |
| 2. O společnosti | 9 |
| 3. Microsoft Visual Studio | 10 |
| 4. Informační systém pro záchrannou službu | 11 |
| 4.1 Komponenty informačního systému | 12 |
| 4.2 Externí technologie..... | 14 |
| 4.3 Komunikace | 15 |
| 4.4 Infrastruktura | 17 |
| 5. Instalační utilita pro update verzí komponent | 18 |
| 5.1 Client update service | 19 |
| 5.2 Update service..... | 21 |
| 5.3 Update Gui tool | 22 |
| 6. Závěr | 24 |
| 7. Seznam často používaných zkratk | 25 |
| 8. Seznam obrázků | 26 |
| 9. Seznam externích zdrojů | 27 |

1. Úvod

Prvním úkolem na začátku stáže bylo seznámení s aktuálním projektem řešeným firmou Vítkovice IT Solutions, to znamená práce s informačním systémem pro zdravotnickou záchrannou službu. Už od začátku stáže mi byla do hlavy vštěpována myšlenka o kvalitě práce a já dobře věděl proč, pracoval jsem na systému, který měl pomáhat při záchraně lidských životů a to 24 hodin denně, 7 dní v týdnu. Kvalitou momentálně myslím spolehlivost, rychlost práce služeb a preciznost při plnění úkolů, které mi byly zadány. S architekturou systému, která byla pro budoucí práci velmi důležitá, jsem se seznámil pomocí dokumentu, který byl vypracován na základě specifikace požadavků ze strany zákazníka.

Po mém úspěšném seznámení s vnitřní strukturou jsem dostal práci, která byla už spíše zaměřena na komponentu klientské strany systému. Komponenta měla za úkol pomáhat klientovi, v mém případě dispečerovi, při vyplňování formuláře o základních informacích volajícího a následné archivace dat. Práce spočívala v úpravě designu formuláře a následného otestování při reálném spuštění.

Po dokončení úkolu jsem byl seznámen s problematikou další práce. Šlo o část systému, která měla za úkol sledovat dostupnost klientských počítačů a v případě potřeby provést update vybrané služby. Tato část systému byla tvořena třemi komponentami - M5UpdateGuiTool, M5Update, M5ClientUpdateService. Popis těchto komponent je dále dokumentu.

Cílem bakalářské stáže je seznámení s prostředím firmy, zabývající se problematikou IT služeb a rozvoj mých získaných znalostí ze školního prostředí pomocí praktického použití v řešení reálného problému. V mém případě rozvoj programování v objektově orientovaném jazyku C#, dotazovacím jazyku (query language) T-SQL a orientace počítačových sítí v řešení problémů systému pro zdravotnickou záchrannou službu.

2. O společnosti

Vítkovice IT Solutions poskytuje komplexní řešení v podobě produktů a služeb na základě požadavků zákazníka v mnoha IT oblastech. Firma se zaměřuje na oblasti řídicích systémů, integrovaných bezpečnostních systémů a systémů pro podporu krizového řízení a plánování. Dalším oborem podnikání Vítkovice IT Solutions jsou ICT systémy pro nově budovanou síť čerpacích stanic pro stlačený zemní plyn - CNG, cloudové řešení a služby agentových aplikací a testování cestovních dokladů s biometrickými prvky.

Mezi významné projekty patří systém elektronického odbavení cestujících na Letišti Václava Havla Praha, tzv. EasyGo, který se stal i IT projektem roku 2012. Šlo o systém, který kontroloval cestující pomocí biometrických pasů.

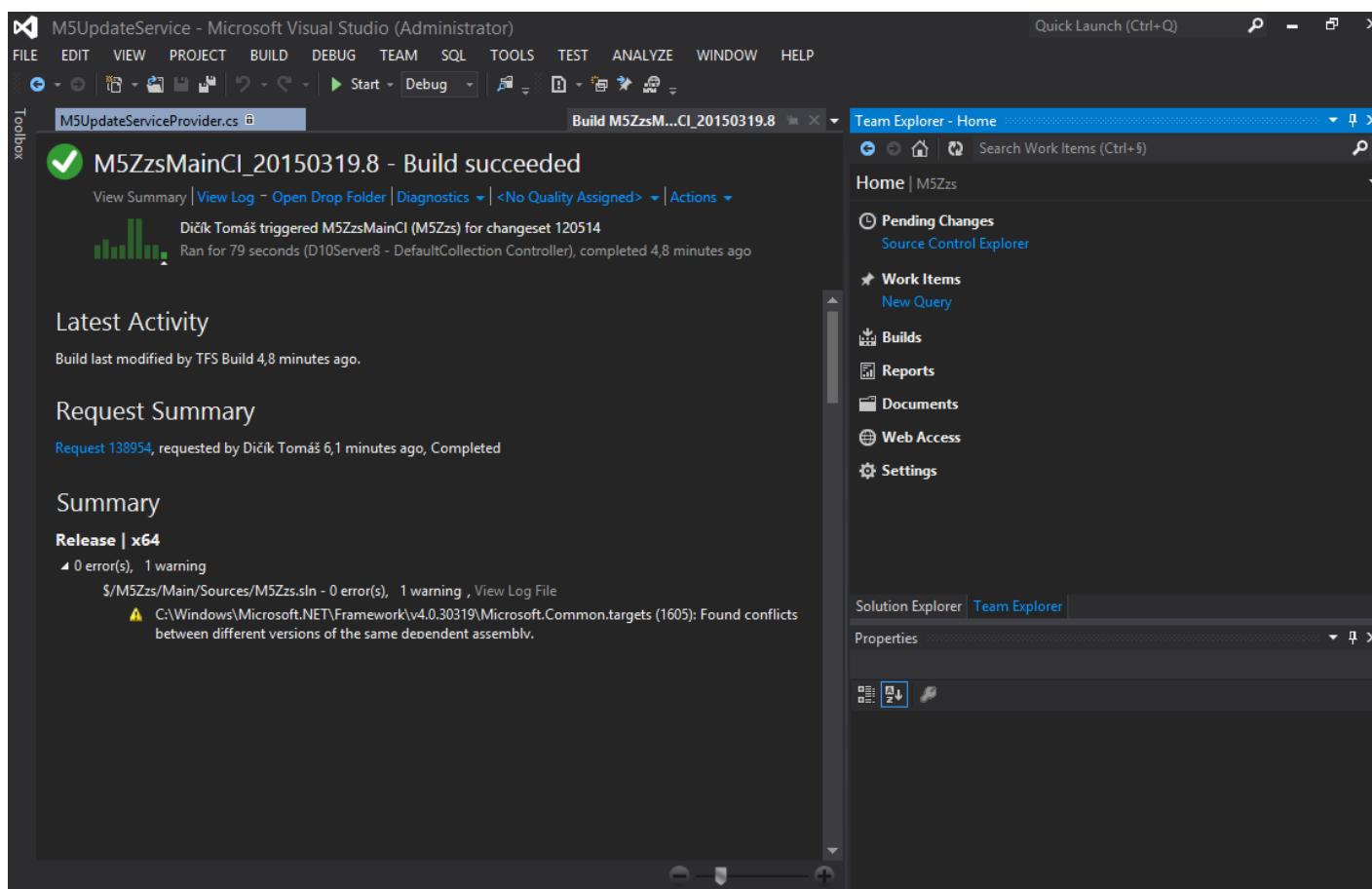
3. Microsoft Visual Studio

Jako vývojové prostředí bylo vývojáři používáno Microsoft Visual Studio. Prostředí podle mě bylo výbornou volbou, protože nabízelo dostatečné množství nástrojů, které firmě celkově ulehčovaly práci na vývoji informačního systému.

Team Explorer - Nástroj výrazně urychlil týmový pokrok ve vývoji softwaru a to hlavně kvůli paralelnímu přístupu více vývojářů k jedné části projektu. Hlavní součástí je TFS (Team Foundation Server), který zprostředkovává v první řadě úložiště pro skripty, které jsou momentálně týmem vytvářeny. V druhé řadě verzování, které umožňuje správu změn, které byly v práci provedeny.

Data Explorer - Nástroj celkově zlepšoval práci s úpravou nebo vytvářením transakcí, které byly zasílány na databázový server, protože bylo Transact SQL plně nástrojem podporováno. Dále nemusel mít vývojář otevřeno zvlášť vývojové prostředí pro práci se C# a T-SQL skripty, proto výrazně snížil nároky na operační paměť PC.

WinForms Designer - Nástroj obsahoval paletu ovládacích prvků s možností úchopu a umístění na povrch rozpracovaného formuláře, který byl zobrazen ještě před spuštěním dané Windows Form, tímto nástroj vývojáře ušetřil psaní kódu a opětovného spouštění celého projektu kvůli kontrole provedených změn designu.



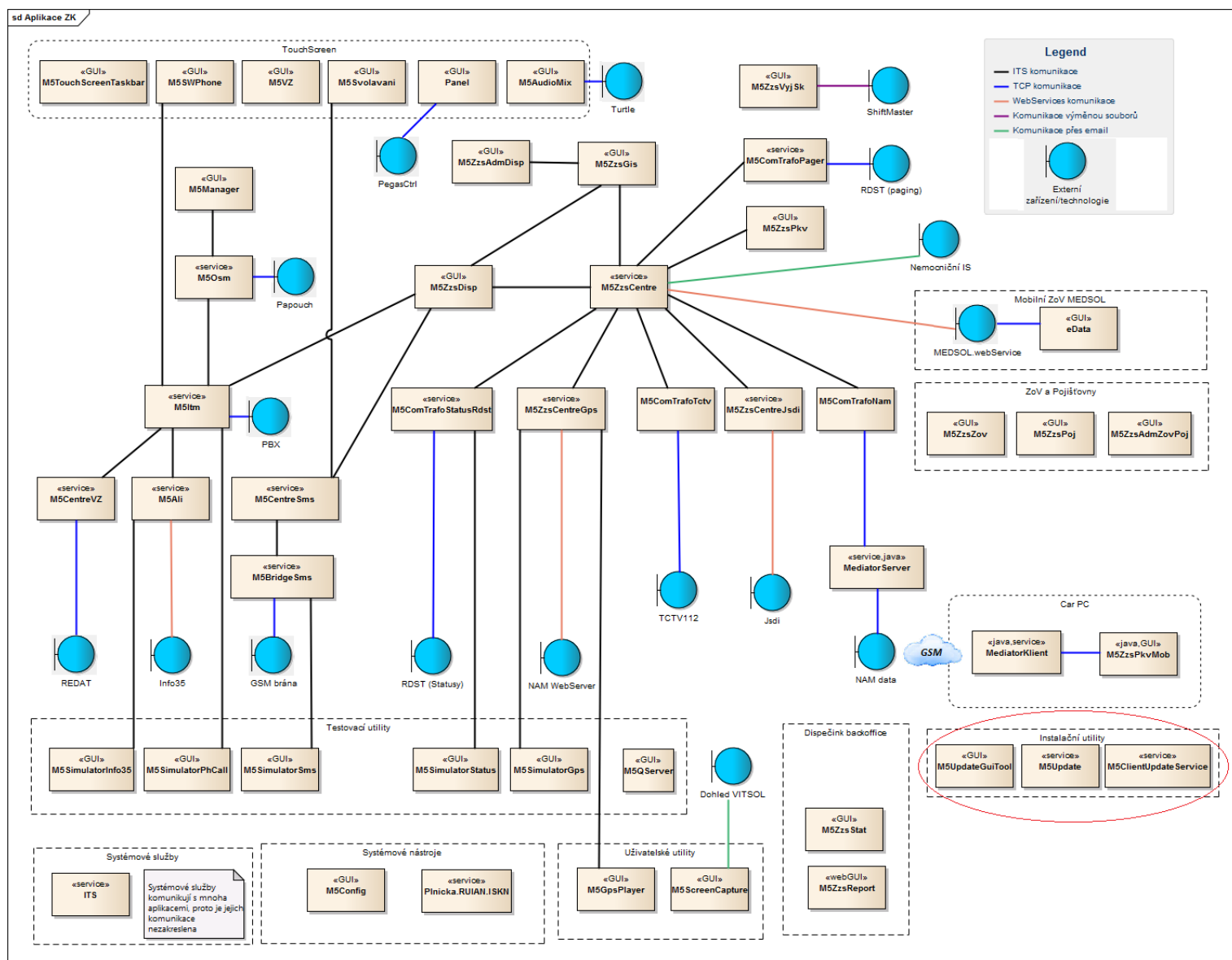
Obrázek 1 : Ukázka testování a následné uložení buildu na Team Foundation Server

4. Informační systém pro záchrannou službu

Schéma popisuje celkovou strukturu informačního systému, která se stará o vytváření, získání, zaslání, zpracování, uložení a následné zobrazení dat uživateli. Podle logické funkčnosti jsou jednotlivé prvky geometricky odlišeny.

Prvky v UML komponent schématu :

- Komponenty informačního systému (obdélník).
- Komunikace (přímka).
- Externí technologie (modrý kruh).



Obrázek 2 : Schéma informačního systému

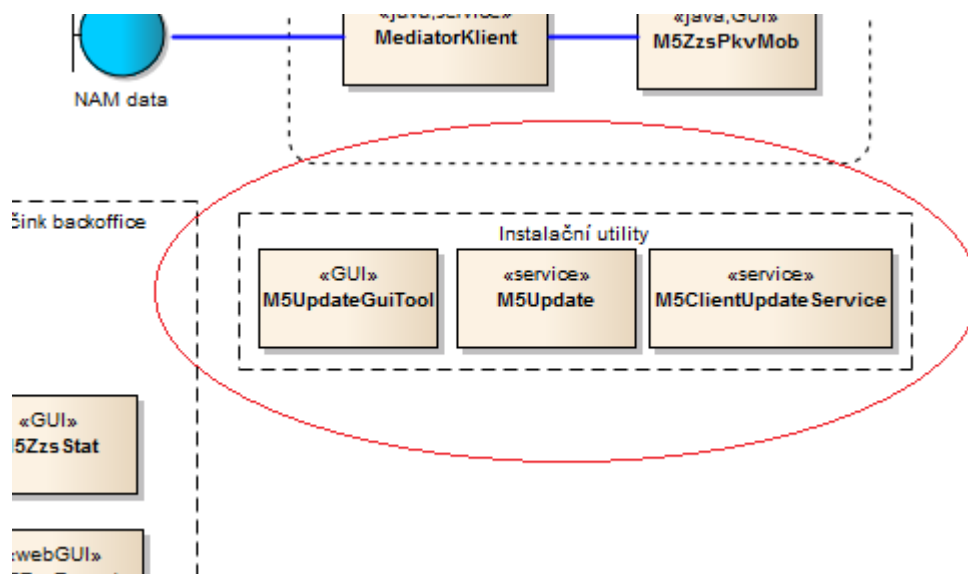
4.1 Komponenty informačního systému

Komponenty jsou podle funkce rozděleny do skupin. První tři jsou součástí aplikační a datové vrstvy, to znamená, že jsou takovým mozkiem celého informačního systému. Zbylé komponenty jsou většinou instalovány na klientských stanicích a prezentují vrstvu prezenční. Skupiny jsou popsány v jednotlivých odstavcích.

Komponenta ZzsCentre je jádrem celého informačního systému. Služba je instalovaná na aplikačním serveru a má na starost řízení komunikace a ukládání dat na databázový server, který je součástí serverového jádra. K jádru jsou na aplikační server dále instalovány komponenty fungující jako takzvaný prostředník, které se starají o získání, zpracování a zaslání dat do jádra, kde jsou následně uloženy. Jako příklad bych mohl uvést službu CentreGPS, která zpracuje GPS data a přepoše je do ZzsCentre jádra, kde se uloží do databáze.

Společná konfigurace a komunikace jsou vlastnosti nezbytné pro správnou funkci celého systému. Komponenta M5config má na starost právě tuto konfiguraci. Sama o sobě je GUI aplikací s vazbou na textový soubor ovlivňující nastavení celého informačního systému. Ke komponentě mají přístup jen kvalifikovaní zaměstnanci spravující stav celého systému. Samotnou mezipřiklační komunikaci implementuje ITS služba popsána v následující kapitole *komunikace*.

Poslední skupinou ovlivňující aplikační vrstvu systému je instalační utilita UpdateService. Utilita řeší problém aktualizace verzí jednotlivých komponent. S utilitou budou opět pracovat jenom kvalifikovaní zaměstnanci starající se o aktualizaci systému. Detailní popis se nachází v samostatné kapitole o této utilitě, kterou jsem měl právě na starost já.



Obrázek 3 : Výřez ze schématu, kde je znázorněna instalační utilita UpdateService

Dále jsou ve schématu komponenty prezentující uživateli data. GUI komponenty mají za úkol jenom převzít připravené data a správně je zobrazit uživateli. Rozdělení komponent není provedeno podle logické funkčnosti, ale podle uživatele, který s jednotlivým GUI pracuje. Uživateli mám na mysli zaměstnance záchranné služby, pro které byl tento informační systém vyvíjen. Design jednoho uživatelského rozhraní jsem upravoval na začátku stáže i já osobně, zde je ukázka z reálného provozu.

Call Taker ZZS - Detail

Výzva

Předání

Nahlášení

Založil

Tester, Test7

Disp.

(Neznámý)

Č. udál.

NP 00002

Co se stalo

BOLEST BŘICHA

Popis:

01:23

Souhrn

Naléhav

Místo události

Oblast:

Střední Morava

Kraj:

Zlínský kraj

Okres:

Obec:

Část obce:

Ulice:

Č. dom/or.:

Patro:

č. bytu:

ZSJ:

Pomíst. n.:

Typ obj.:

Objekt:

N:

49°13'36"

E:

17°39'59"

Typ souř.:

WGS84 (ssmmvv)

Význ. souř.:

Kraj

Smazat celou adresu

Poznámka k místu události:

Okolní události

Událost

Výzva

Souhrn

Místo událo

Výzvy

Klasifikace

Strom klasifikací

Směr

Typ

Tel. číslo

Kdo volá

Stav zpracování

725499393

Převzato

Převzít místo

Přidat

Odebrat

Směr

Příchozí

Telefonní číslo

725499393

Kdo volá

Stav zpracování

Převzato

Čas výzvy

11.03.2015 11:03:57 Z

Číslo výzvy

☎ Telefon

725499393

☒ Datová věta

☒ Adresa

,

Čas

Krok

Stav

Příjmení

Jméno

Rod.č./věk

Poznámka

TANR

TAPP

First Responder

Historie pacienta

TYP

Výzva

Souhrn

Diagnóza

Zpús. řešení

Zpráva kont. místo

NACA

Odhad.příjezd (min.)

Prac. diag.

Kont.m.

Směrování

Zpráva od.

přijata

ZOV

Způsob řešení

Og

Zdrav. zařízení

Předán kdy

komu

Oddělení

Požadavky

DNR

0

0

0

0

0

0

0

0

- Neurčeno -

ZZS

MěP

HZS

LZS

PČR

HSL

TCTV

MLUVČÍ

Místopis:

Příliš krátký text

Hledáno (v DB): 0 (0)

Nalezeno: 0

Zobrazit v mapě

Obrázek 4 : Ukázka dispečerské, mnou upravované, GUI komponenty

4.2 Externí technologie

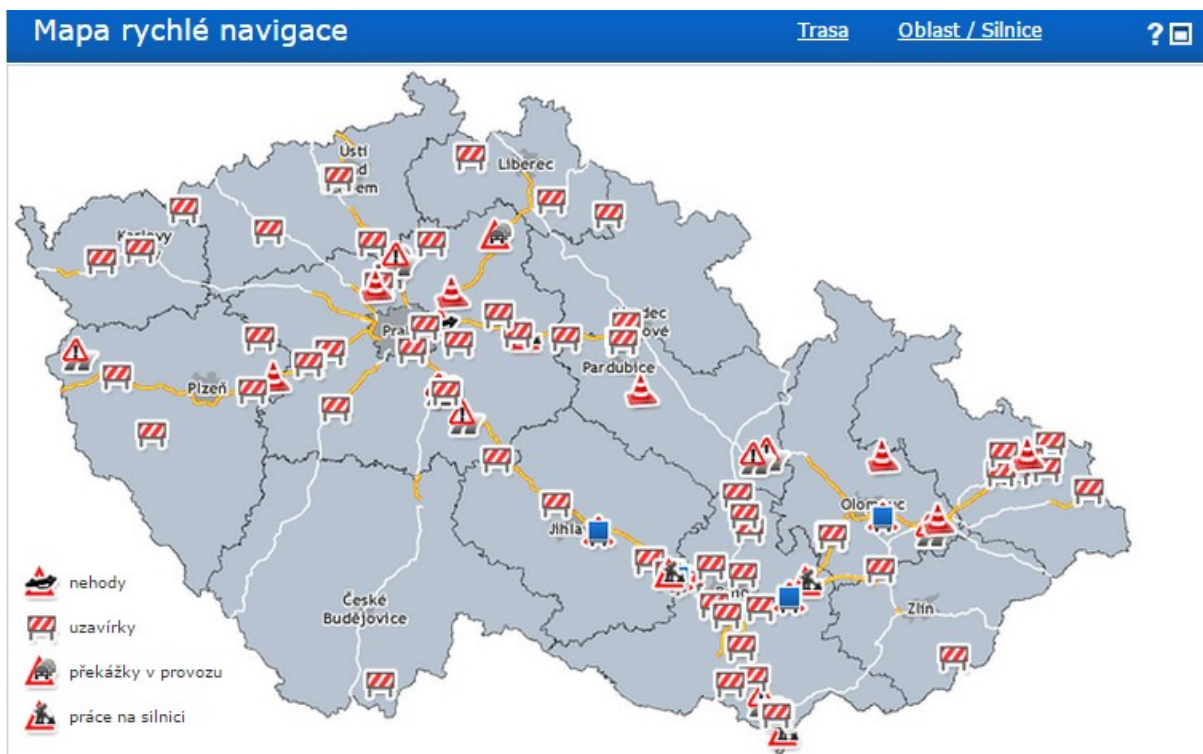
Nedílnou součástí systému jsou i technologie provozované nějakou třetí stranou. Ve schématu lze funkcionálně odlišit dvě skupiny těchto technologií.

Rozdělení :

- Koncová zařízení.
- Informační systémy.

Koncová zařízení - Koncová zařízení mají za úkol převzít softwarová data a nějakým způsobem ovlivnit fyzickou vrstvu. Fyzickou vrstvou je myšleno úplné koncové zařízení, které je elektronicky ovládáno. Popíši příklady, kdy budou softwarovými daty myšleny stavy jednotlivých dispečerů a následné chování koncových zařízení. Například takové zařízení Papouch elektronicky ovládá světelné majáky prezentující dané stavy dispečerů v závislosti na změně dat. Stejný zdroj dat používá také koncové zařízení Turtle, které je vlastně zvukovým směšovačem audio zařízení daných dispečerů. Koncová zařízení v konečném důsledku ušetří zbytečnou práci, která je vykonávána automaticky, tím výrazně zefektivní dispečerům práci.

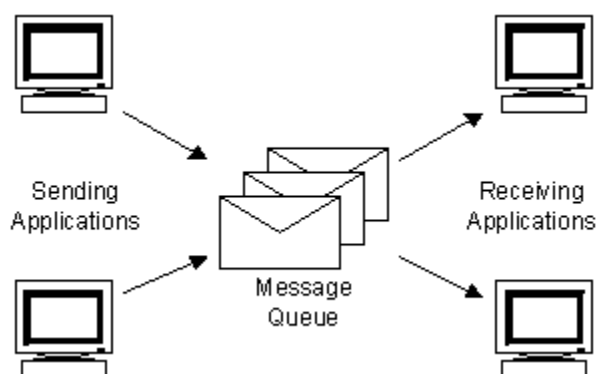
Informační systémy - Externí informační systémy jsou využity k uchování a zaslání nějakého druhu dat, které jsou potřebné v systému záchranné služby. Druh dat je od daného externího informačního systému velmi odlišný. Externí systém Info35 například poskytuje data o aktuálním volajícím, kdežto takový systém JSDI pomáhá při výběru správné cesty pro transport raněného, protože poskytuje data o nehodách a omezeních na okolních pozemních komunikacích.



Obrázek 5 : Ukázka webové prezentace informačního systému JSDI

4.3 Komunikace

ITS komunikace (IZS (integrovaná záchranná služba) talk server) – Slouží pro komunikaci mezi komponentami informačního systému. Komunikaci realizuje ITS server, ve kterém probíhá frontový FIFO (first in first out) proces ukládání zpráv. Samotné zasílání zpráv funguje na technologii Microsoft MSMQ (Message Queuing). Data, které jsou posílány ITS komunikací, jsou samy o sobě ve formátu XML (Extensible Markup Language). Technologie nám poskytuje velmi důležité vlastnosti pro komunikaci jako je například bezpečný, rychlý přenos zpráv a spolehlivost v případě dočasného výpadku. Během výpadku jsou zprávy udrženy ve frontě, ze které jsou po obnovení komunikace opět zaslány ve správném pořadí do komponent, kde měly být zaslány před výpadkem.



Obrázek 6 : Schematické znázornění MSMQ komunikace

TCP komunikace (Transmission Control Protocol) – Soketová TCP komunikace poskytuje spolehlivý obousměrný přenos sekvenčních dat zajišťující doručení ve správném pořadí. Skvělá volba pro nepřetržité zasílání řídicích zpráv pro danou externí jednotku. Navíc komunikace nabízí v případě výpadku rychlou detekci přerušení kanálu, což je v našem případě klíčové. Při bližším pohledu na komunikaci mezi M5Itm a externí technologií PBX (Private branch exchange) lze vyzorovat využití výše zmíněných výhod. Technologie PBX je zařízení sjednocující výstupní body telefonních přístrojů do veřejné telefonní sítě. Další funkcí zařízení je i ovládání linek. Služba M5Itm zasílá řídicí signály pro PBX ovlivňující celou telefonii, u které vlastně monitorujeme i síťovou dostupnost kvůli soket detekce přerušení kanálů.

WebServices komunikace - V systému je použita vždy v případě, kdy potřebujeme dostat nějaké větší množství dat, získané z externího zdroje, do služby v našem systému. Data se posílají prostřednictvím protokolu SOAP (simple object access protocol), takže podporuje přenos serializovaných dat ve formátu XML. Skvělým příkladem je lokální služba M5Ali, která zpracovává data z externího zdroje Info35 poskytující data o volajícím účastníkovi události.

GSM (Globální Systém pro Mobilní komunikaci) – Jediný bezdrátový typ komunikace v celém systému. Je použita mezi mobilní komponentou CarPC, která je instalována v každém vozidle zdravotní služby a informačním systémem. Komponenta poskytuje odeslání dat o aktuální pozici a stavu vozidla mobilní GSM síti. Data se následně pomocí technologie NAM (The Cisco Network Analysis Module) Traffic Analyzer zpracují a posílají dále do systému, kde se s daty pracuje.

Email komunikace - Po výjezdu a rychlém ošetření pacienta je potřeba dovést postiženého do nejbližší nemocnice a provést důkladnější vyšetření. Rychlost přípravy zaměstnanců nemocnice může být klíčová, proto každý vůz záchranné služby informuje nemocnici o daném stavu pacienta, pozici a předpokládaném příjezdu do nejbližší stanice pohotovosti. Je zbytečné každou nemocnici nutit vytvářet nové technologie pro komunikaci s naším systémem, proto zasílání zpráv do dané nemocnice funguje na principu elektronické pošty. SMTP server uskutečňující přenos e-mail zpráv je umístěn v serverovně společně s ostatními serverovými jednotkami realizující funkcionalitu celého systému.

FTP komunikace (File Transfer Protocol) - Typ FTP komunikace je zřízena mezi komponentou ZzsVyj_Sk vytvářející skupiny záchranářů účastnících se daného výjezdu a externí technologií ShiftMaster. Technologie ShiftMaster pracuje s daty zaměstnanců, v našem případě záchranářů a poskytuje data o jejich směnách. Výsledné data jsou předávány formou textového souboru, proto je pro komunikaci zvolen FTP přenos.

The screenshot shows the 'reálný plán na rok 2010 pro pracoviště 50101 Děts.neurol.kl.-st.l.-menší děti' (real plan for 2010 for workplace 50101 Children's neurology clinic - small children). The interface includes a calendar for January 2010, a list of employees, and a detailed shift schedule table. A red box highlights the 'Plán a záznamy' (Plan and Records) section.

| Zaměstnanec | Minul. saldo | Odprac./neodprac./prop | Měs. fond | Saldo | Saldo vyr.obd. | 31. čt | 1. pá | 2. so | 3. ne | 4. po | 5. út | 6. st | 7. čt | 8. pá | 9. so | 10. ne | 11. po | 12. út | 13. st |
|-------------------|--------------|------------------------|-----------|--------|----------------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|
| Bystroňová Katka | (0) | +132/0/0 | 143 | (-11) | (-737) | | | | | | | | R12 | R12 | | | | | |
| Dyloková Marie | (0) | +162/0/0 | 151 | (+11) | (-705,5) | | | | | N12 | | N2+ | N2+ | N2+ | | | | | |
| Kabišková Martina | (0) | +174/0/0 | 0 | (+174) | (+162) | | N2+ | | N2+ | N2+ | N2+ | | | R12 | | | | | |

Obrázek 7 : Ukázka práce s externím informačním systémem Shiftmaster

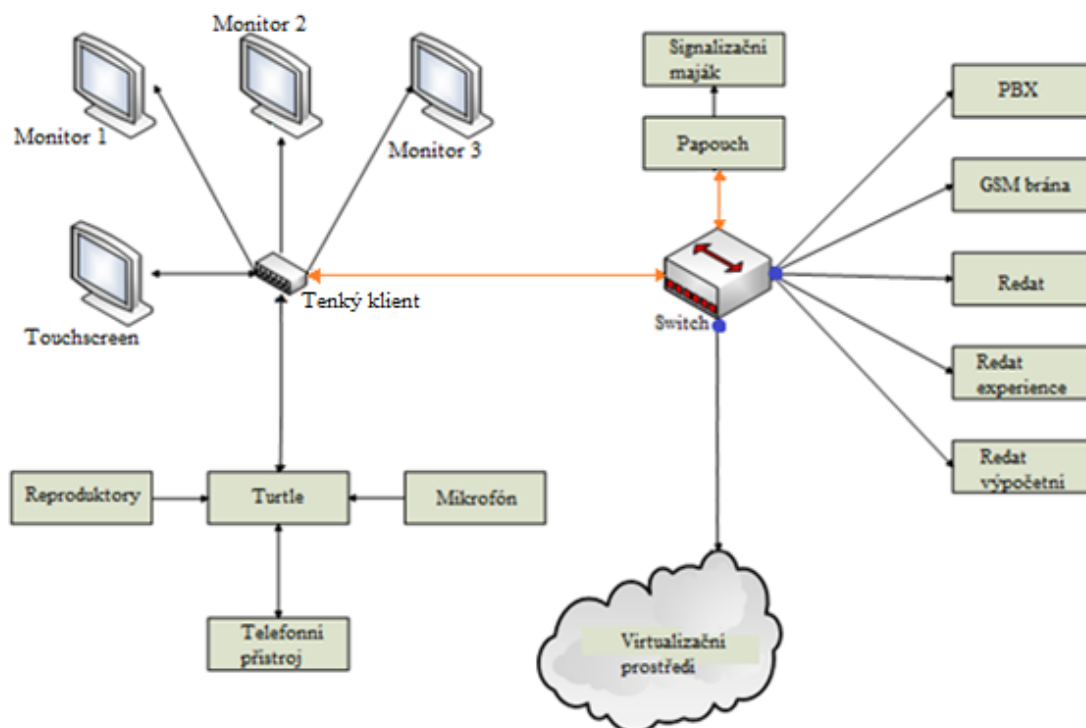
4.4 Infrastruktura

Informační systém je fyzicky složen ze serverového jádra, operátorských pracovišť, pracovišť na výjezdových stanovištích a mobilních zařízení, instalovaných ve vozidlech záchranné služby. Schéma znázorňuje infrastrukturu serverové a dispečerské části. Switch (přepínač) nám rozděluje právě tyto části a řídí síťovou komunikaci mezi nimi.

Části fyzicky umístěné v serverovém jádru jsou znázorněny modrým bodem v připojení s přepínačem. Oranžový ukazatel kontaktu s přepínačem označuje zase místa umístěné už přímo někde na pracovištích. Samotné pracoviště dispečerů jsou tenkými klienty.

"Tenký klient (překlad anglického thin client) je počítač nebo počítačový program, který při plnění svého úkolu silně závisí na jiném počítači (na svém serveru)."

V našem případě funkce dispečerských počítačů závisí na serverech, které jsou hardwarovými i aplikačními virtuálními stroji. Konstrukce datového centra, ve kterém jsou umístěny servery, nabízí ochranu před ohněm, teplem, vodou, kouřem, prachem explozí, troskami, elektromagnetickým rušením, vandalismem, neautorizovaným přístupem.



Obrázek 8 : Schéma infrastruktury

Virtuální stroje jsou ke každému operátorskému fyzickému stroji přiřazeny s ohledem na optimální rozložení zátěže mezi jednotlivé servery realizující virtualizaci. Návrh zahrnoval i eliminaci chyb způsobené případným výpadkem některého ze serverů.

Součástí serverového jádra je i zařízení Redat, umožňující nahrávání telefonních hovorů operátorů a následná funkce rozpoznání hlasu, která je implementována v softwarové komponentě Redat experience. Samotné výpočty hlasové analýzy probíhají na Redat výpočetním serveru.

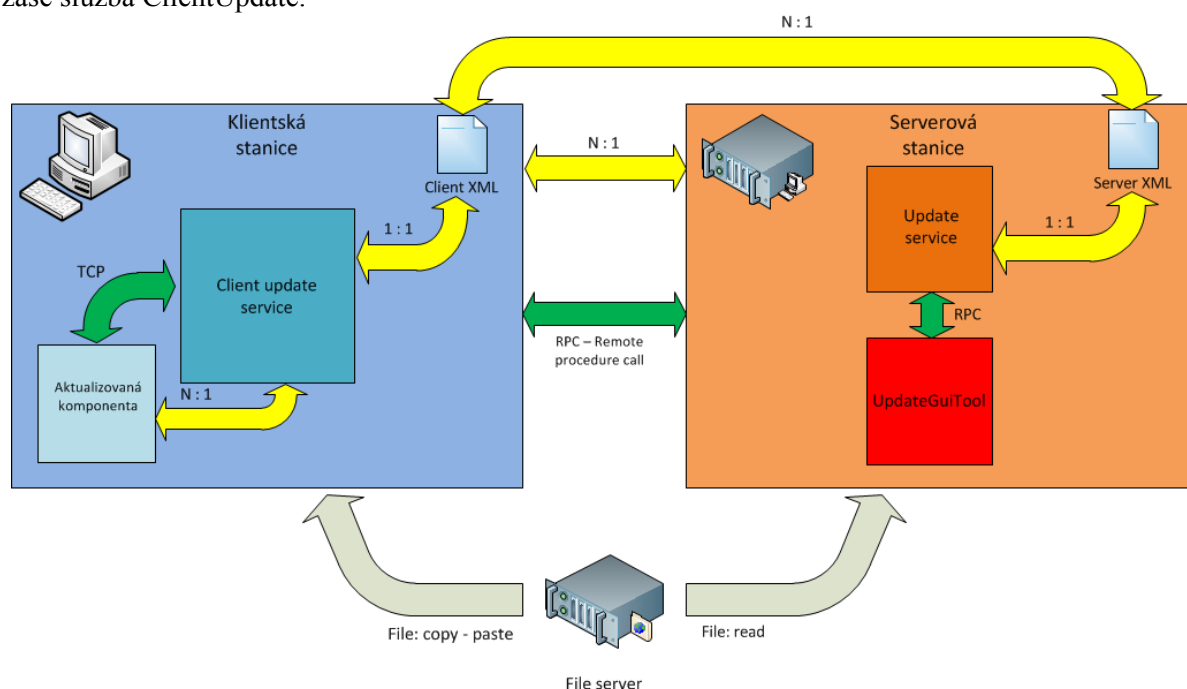
GSM brána slouží pro příjem a rozesílání tísňových SMS zpráv mezi vozy záchranné služby a operátory. Detaily jsou uvedeny v článku věnující se komunikaci.

Jak lze vidět, koncová audio zařízení jsou napojena do zvukového směšovače Turtle, který jsem už popisoval v odstavci věnující se koncovým zařízením. Totéž platí pro signalizační maják, ovládaný zařízením Papouch.

5. Instalační utilita pro update verzí komponent

Schéma obsahuje základní prvky důležité pro celkovou funkci instalační utility pro aktualizaci komponent. Fyzicky funkcionalitu zajišťují tři hlavní části, server (teplé barvy), klientská stanice (studené barvy) a samostatný souborový server (umístěn uprostřed dole).

Aplikační logiku serverové části implementuje služba UpdateService, která je ovládaná pomocí UpdateGuiTool příslušným zaměstnancem starajícím se o aktualizace. U klienta řídí aplikační logiku zase služba ClientUpdate.



Obrázek 9 : Schéma celkové struktury aktualizací utility

Popis vztahů

"Vztah je obecná vlastnost konkrétního objektu nebo subjektu (entity), která se váže k jinému objektu nebo subjektu (entitě). Jedna entita může mít více různých vazeb k vícero různým entitám, vzájemné vztahy těchto entit mohou v praxi někdy tvořit velmi rozsáhlé sítě a vzájemné propletence."

Klientská stanice - serverová stanice

Relační vztah mezi těmito entitami je N : 1, kde N je počet dispečerských stanic.

Klientská stanice - aktualizovaná komponenta

Relační vztah mezi těmito entitami je 1 : N, kde N je počet komponent na dané stanici. Počet komponent je na každé stanici odlišný.

Klientská stanice - XML klient data

Relační vztah mezi těmito entitami je 1 : 1.

Serverová stanice - XML server data

Relační vztah mezi těmito entitami je 1 : 1.

XML klient data - XML server data

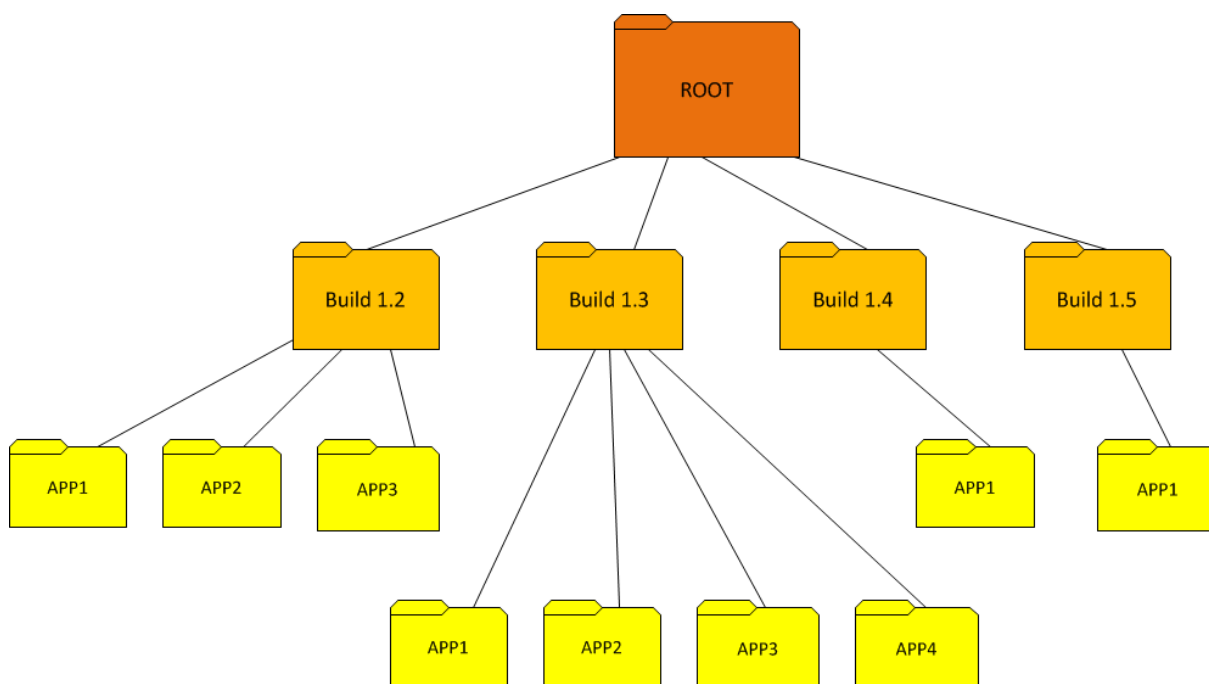
Relační vztah mezi těmito entitami je N : 1. Kde N je počet XML dat z jednotlivých stanic v server XML.

File server

Souborový server obsahuje nové verze daných komponent. Obsah dat na serveru mění vývojáři, kteří vytváří tyto verze. Právě team foundation server, popsán v kapitole visual studio, má na starost přímou změnu obsahu.

Popis souborové struktury

Datová struktura je uspořádána do stromu, kde je kořenem složka, která má své specifické souborové oprávnění. Síťově je dostupná jen pro vývojáře a instalační utilitu, která může ze složky jenom číst. Uzel kořenu jsou složky pojmenované dle daných verzí. Uzel verze obsahuje listy pojmenované dle daných komponent. Listy už obsahují jen soubory potřebné pro aktualizaci zvolené komponenty.



Obrázek 10 : Ukázka stromové struktury souborového serveru

5.1 Client update service

Provoz započne ihned po zapnutí klientské stanice, služba běží na pozadí.

Seznam činností:

- Kopírování souborů ze síťové složky na příkaz serveru.
- Vypnutí aktualizované komponenty v případě instalace.
- Instalace nové komponenty.
- Aktualizace klientského XML souboru.
- Cycklické ohlašování klienta o své přítomnosti.

Kopírování souborů ze síťové složky na příkaz serveru

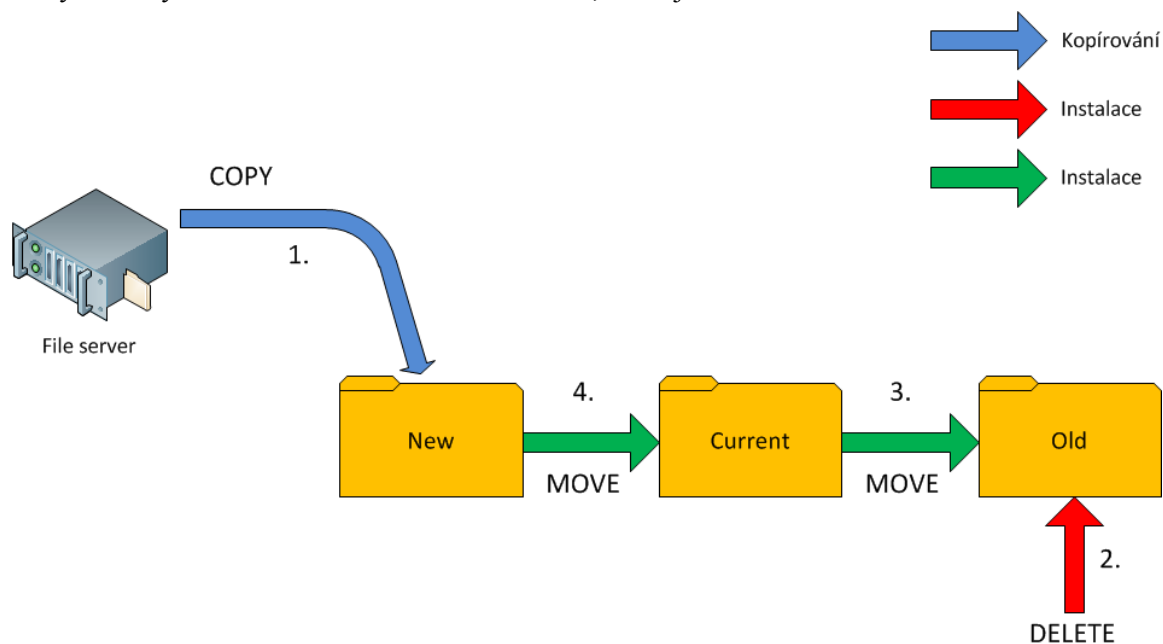
Aktualizace komponenty v klientském PC započne kopírováním instalačních souborů z file serveru na dočasné lokální úložiště stanice. Služba obsahuje soubor config.txt nesoucí data o cestě k úložišti a instalační složce. Config každý klient konfiguruje při prvním spuštění služby. Součástí kopírování je i rozbalení instalačního souboru typu .rar a jeho následné smazání.

Vypnutí aktualizované komponenty v případě instalace

Protože se při instalaci pracuje se soubory aktualizované komponenty, je nutné provoz této komponenty ukončit. Komunikace mezi službou a komponentou je realizován pomocí TCP pipelines spojení. V momentu požadavku o ukončení má uživatel možnost tento požadavek odložit, ale jen dočasně.

Instalace nové komponenty

Soubor, kde je instalována komponenta obsahuje 3 hlavní složky jménem current, new a old. Ve složce current jsou soubory potřebné pro provoz komponenty, takže komponenta právě pracuje s těmito soubory. Dočasným úložištěm jsem v předchozím kroku kopírování myslel složku new, kde jsou teď umístěny soubory čekající na instalaci. Ta je prováděna velmi jednoduše, služba přesune všechny soubory z current do old a z new do current, tímto je instalace u konce.



Obrázek 11 : Ukázka provádění operací při kopírování a instalaci

Aktualizace klientského XML souboru

Každá klientská stanice nese svůj vlastní XML soubor obsahující data o svém počítači a nainstalovaných komponentách. Data se aktualizují buď při spuštění služby a nebo po instalaci. Součástí funkcionality služby je i rozpoznání manuální modifikace dat a v případě nekorektnosti jejich napravení.

Ukázka:

```
<clientinfo stationname="test_station_name">
  <appmodule name="M5ZzsDisp" version="1.2" path="/moduly/M5ZzsDisp"></appmodule>
  <appmodule name="M5Itm" version="1.4" path="/moduly/M5Itm"></appmodule>
</clientinfo>
```

Cyklické ohlašování klienta o své přítomnosti

V moment požadavku o ohlášení daného klienta je nutné kontaktovat server. Komunikace mezi klientem a serverem probíhá pomocí vzdáleného volání procedur (RPC - remote procedure call). V parametrech procedury jsou klienty vždy zaslány XML data obsahující všechny potřebné informace.

5.2 Update service

Provoz započne ihned po zapnutí serverové stanice, služba běží na pozadí a měla by běžet nepřetržitě.

Seznam činností :

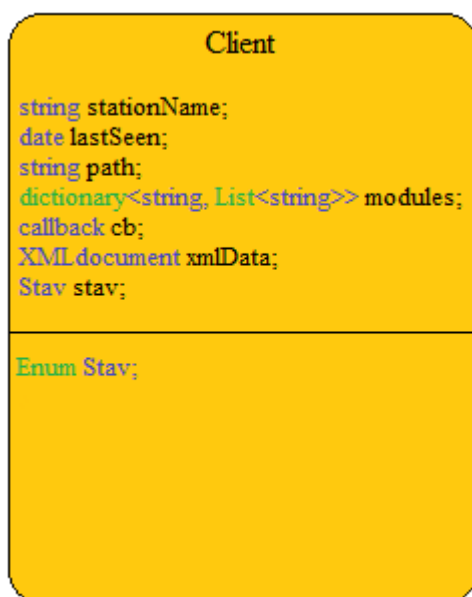
- Funkce prostředníka v komunikaci mezi klienty a GUI.
- Cyklická kontrola dostupnosti klientů.
- Ukládání a úprava serverových XML dat o klientech.

Funkce prostředníka v komunikaci mezi klienty a GUI

Předlohou konstrukce této služby byl návrhový vzor ¹Mediator (prostředník). Výhodou návrhového vzoru je v našem případě oddělení prezentační a aplikační vrstvy, takže není stanice, která prezentuje data zbytečně zatěžována výpočty. Komunikace probíhá opět pomocí vzdáleného volání procedur (RPC - remote procedure call), kde jsou potřebné data zaslány v parametrech procedury.

Cyklická kontrola dostupnosti klientů

Služba obsahuje třídu s atributy prezentující data klienta a jeho callback což je v RPC komunikaci ukazatel na funkci, která umožňuje daného klienta kontaktovat zpět. Po startu klientské služby se klient ohlásí pomocí zaslání svých XML dat, které se na straně serveru zpracují a uloží spolu s callback ukazatelem do objektu prezentujícího daného klienta. Princip kontroly dostupnosti spočívá v pravidelném průchodu klientských objektů a pomocí callbacku každého požádat o své ohlášení. V případě síťové nedostupnosti se objekt klienta smaže a kontaktuje se GUI.



Obrázek 12 : Vizualizace třídy klienta

Ukládání a úprava serverových XML dat o klientech

Důvod, proč na serveru uchováváme data o jednotlivých klientech, je jejich následná prezentace v GUI. Proto je u XML dat kladen důraz na jejich věrohodnost. Data se automaticky aktualizují v procesu ověření přítomnosti klientů. V případě prvního spatření klienta jsou data zpracovány a přidány k existujícím serverovým datům. Po aktualizaci se data pošlou GUI, které je opět prezentuje. Při smazání nebo poškození dat server změnu rozpozná a je schopen vrátit data zpět do původního stavu.

¹ Znalost získána z předmětu Vývoj informačních systémů

5.3 Update Gui tool

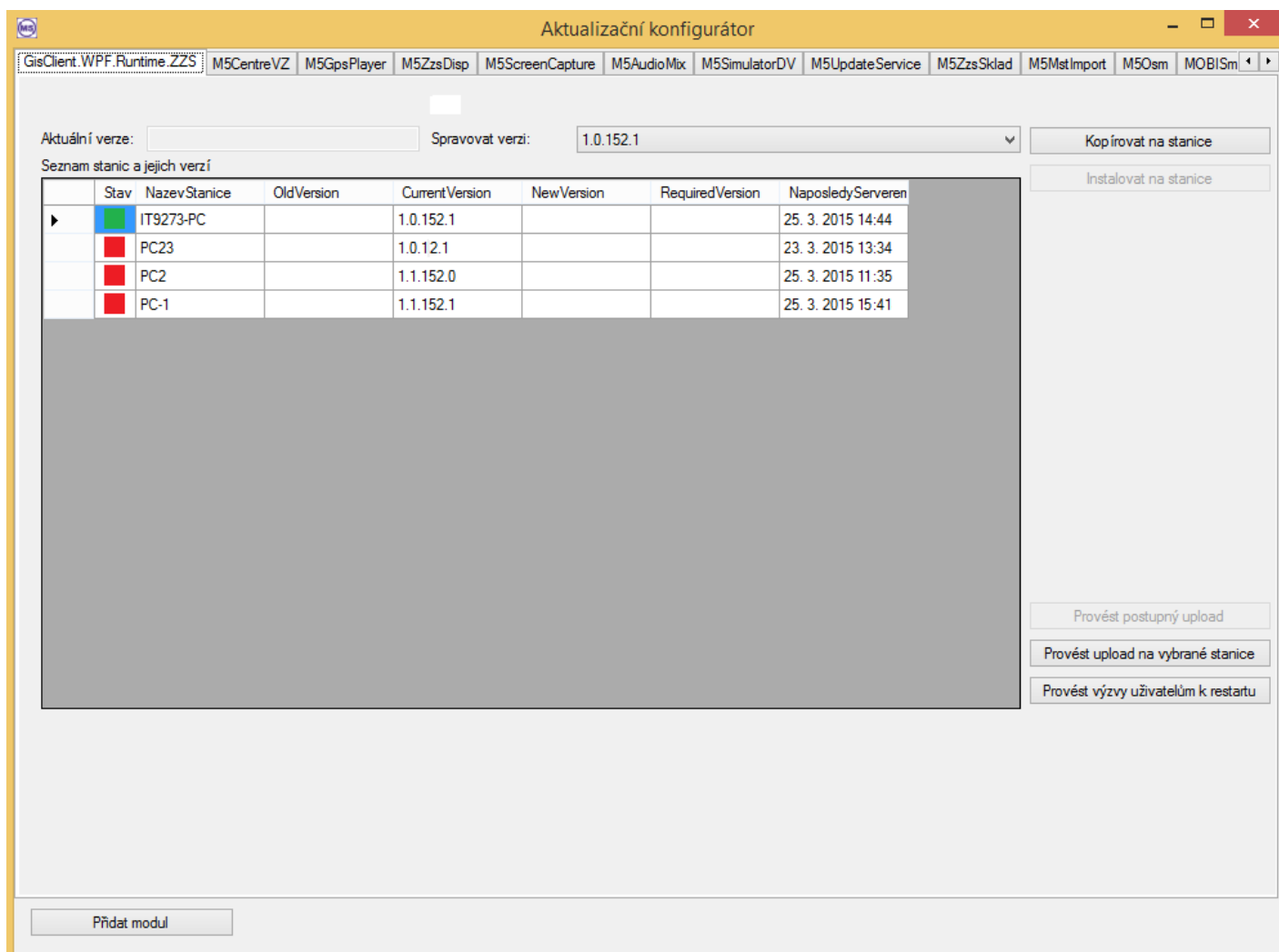
Gui komponenta je pro uživatele jediným způsobem jak ovládat celý systém aktualizace komponent, její implementace byla provedena tak, aby komunikace mezi klientem a serverem fungovala stále na pozadí.

Seznam činností :

- Prezentace klientských XML dat
- Čtení dat ze souborového serveru
- Kontrola důvěryhodnosti instalačních souborů

Prezentace klientských XML dat

Prezentace zahrnuje rozparsování XML dat a jejich transformace do objektů, které dokáže tabulka zobrazit. Data jsou zobrazena pomocí ovládacího prvku GridView, který jako zdroj dat používá svůj objekt dataSource, kterému předáváme naše zpracované XML data. Každá záložka, prezentující jeden modul komponenty, používá svůj vlastní GridView s rozdílným zdrojem dat.



Obrázek 13 : Ukázka uživatelského rozhraní aktualizační utility

Čtení dat ze souborového serveru

Záložky se vytváří pomocí třídy `TabPage`, kde se jednotlivé objekty této třídy vytváří v závislosti na počtu složek modulů na souborovém serveru. Jednotlivý objekt `TabPage` obsahuje právě zmíněný `gridView` a další ovládací prvky jako tlačítka nebo listboxy obsahující seznam verzí, na které můžeme danou komponentu aktualizovat. Seznam dostupných verzí získáváme také čtením dat ze souborového serveru.

Kontrola důvěryhodnosti instalačních souborů

V RPC komunikaci mezi serverem a klientem je před kopírováním v parametru také poslán checksum třídy MD5. Vygenerovaný hash je datového typu `string` a je získán z instalačního souboru formátu `.zip`. Na straně klienta se hash vygeneruje také, teď už ale z okopírovaného instalačního souboru a v posledním kroku se hashe porovnají. Protože se hash tvoří z binárních dat souborů, můžeme tímto způsobem zkontrolovat případné poškození během síťového kopírování.

6. Závěr

Hlavním cílem bakalářské praxe bylo praktické využití teoretických znalostí ze školního prostředí a seznámení s pracovním prostředím IT firmy.

Postupem času rostla jak složitost úkolů, které mi byly zadávány, tak i počet předmětů, které jsem musel v praxi použít. Z počátku to byla práce týkající se úpravy designu, tam jsem využil praktické i teoretické znalosti z předmětu uživatelské rozhraní. Další prací bylo seznámení s celým informačním systémem záchranné služby, tam jsem využil znalosti z předmětů zaměřených na softwarové inženýrství, počítačové sítě a vývoj informačních systémů. Ve zbytku praxe jsem používal hlavně praktické znalosti z předmětů týkajících se implementace v jazyku C# a T – SQL.

Musím přiznat, že jsem až na praxi pochopil váhu získaných školních znalostí a překvapilo mě i zvýšení mého zájmu v předmětech, které mě ve školní lavici moc neoslovily. Tímto bych řekl, že cíl týkající se využití školních znalostí v praxi byl splněn.

Prostřednictvím stáže jsem i poznal jakým způsobem probíhá přidělování a řízení práce jednotlivých skupin IT vývojářů pracujících na daném projektu. S personálem firmy jsem byl nadmíru spokojen. Vyšli mi vždycky vstříc, to jak s pomocí ohledně mé práce, pracovních prostředků, tak i věcí administrativně zaměřených. Tímto bych určil můj druhý cíl také jako splněný.

7. Seznam často používaných zkratek

XML (Extensible Markup Language) - obecný značkovací jazyk. Jazyk je určen především pro výměnu dat mezi aplikacemi a pro publikování dokumentů, u kterých popisuje strukturu z hlediska věcného obsahu jednotlivých částí, nezabývá se vzhledem.

GUI (Graphical User Interface) - je uživatelské rozhraní, které umožňuje ovládat počítač pomocí interaktivních grafických ovládacích prvků.

RPC (Remote Procedure Call) - je technologie dovolující programu vykonat proceduru, která může být uložena na jiném místě než je umístěn sám volající program. Příkladem budiž výpočet funkce na jiném počítači v síti.

RAR (Roshal ARchive) - je proprietární souborový formát pro kompresi dat a archivaci.

ITS (IZS (integrovaná záchranná služba) Talk Server) - je technologie sloužící pro komunikaci mezi komponentami informačního systému.

TCP (Transmission Control Protocol) - je nejpoužívanějším protokolem transportní vrstvy v sadě protokolů TCP/IP používaných v síti Internet.

FTP (File Transfer Protocol) - je v informatice protokol pro přenos souborů mezi počítači pomocí síťové vrstvy.

TFS (Team Foundation Server) - vývojáři sdílený server sloužící pro ukládání vyvíjených skriptů.

C# (C Sharp) - je vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý firmou Microsoft zároveň s platformou .NET Framework.

SQL (Structured Query Language) - standardizovaný strukturovaný dotazovací jazyk, který je používán pro práci s daty v relačních databázích.

T - SQL (Transact SQL) - je proprietární rozšíření do SQL od společností Microsoft.

CNG (Compressed Natural Gas) - je zkratka pro stlačený zemní plyn.

M5 - verze celého informačního systému.

ICT (Information and Communication Technologies) - zahrnují veškeré informační technologie používané pro komunikaci a práci s informacemi.

JSDI (Jednotný Systém Dopravních Informací) - je komplexním systémovým prostředím pro sběr, zpracování, sdílení, distribuci a publikaci dopravních informací a dopravních dat o aktuální dopravní situaci a informací o pozemních komunikacích.

PBX (Private branch exchange) - je zařízení, které sjednocuje výstupní body všech firemních telefonů do veřejné telefonní sítě.

SOAP (Simple Object Access Protocol) - je protokolem pro výměnu zpráv založených na XML přes síť, hlavně pomocí HTTP.

8. Seznam obrázků

| | |
|---|----|
| Obrázek 1 : Ukázka testování a následné uložení buildu na Team Foundation Server..... | 10 |
| Obrázek 2 : Schéma informačního systému | 11 |
| Obrázek 3 : Výřez ze schématu, kde je znázorněna instalační utilita UpdateService | 12 |
| Obrázek 4 : Ukázka dispečerské, mnou upravované, GUI komponenty | 13 |
| Obrázek 5 : Ukázka webové prezentace informačního systému JSDI | 14 |
| Obrázek 6 : Schematické znázornění MSMQ komunikace | 15 |
| Obrázek 7 : Ukázka práce s externím informačním systémem Shiftmaster | 16 |
| Obrázek 8 : Schéma infrastruktury..... | 17 |
| Obrázek 9 : Schéma celkové struktury aktualizací utility | 18 |
| Obrázek 10 : Ukázka stromové struktury souborového serveru..... | 19 |
| Obrázek 11 : Ukázka provádění operací při kopírování a instalaci..... | 20 |
| Obrázek 12 : Vizualizace třídy klienta..... | 21 |
| Obrázek 13 : Ukázka uživatelského rozhraní aktualizací utility | 22 |

9. Seznam externích zdrojů

Obrázek 5 - <http://www.dopravniinfo.cz>

Obrázek 6 - <https://msdn.microsoft.com/en-us/library/ms711472%28v=vs.85%29.aspx>

Obrázek 7 - <http://www.ivar.cz/rs/?cat=12>

Citace - Tenký klient - http://cs.wikipedia.org/wiki/Tenk%C3%BD_klient

Citace - Vztah- <http://cs.wikipedia.org/wiki/Vztah>